The structure of subword graphs and suffix trees of Fibonacci words

Wojciech Rytter[1,2]

[1] Instytut Informatyki, Uniwersytet Warszawski, Banacha 2, 02–097, Warszawa, Poland
[2] Department of Computer Science, New Jersey Institute of Technology
rytter@mimuw.edu.pl. *

Abstract. We use automata-theoretic approach to analyze properties of Fibonacci words. The directed acyclic subword graph (dawg) is a useful deterministic automaton accepting all suffixes of the word. We show that dawg's of Fibonacci words have particularly simple structure. The simple structure of paths in these graphs gives simplified alternative proofs and new interpretation of several known properties of Fibonacci words. The structure of lengths of paths in the compacted subword graph corresponds to a number-theoretic characterization of occurrences of subwords in terms of Zeckendorff Fibonacci number system. Using the structural properties of dawg's it can be easily shown that for a string w we can check if w is a subword of a Fibonacci word in time $O(|w|)$ and $O(1)$ space. Compact dawg's of Fibonacci words show a very regular structure of their suffix trees and show how the suffix tree for the Fibonacci word grows (extending the leaves in a very simple way) into the suffix tree for the next Fibonacci word .

1 Introduction

Fibonacci words form a famous family of words, due to many interesting properties related to text algorithms and combinatorics on words, see [2, 10]. In particular Fibonacci words have $\Theta(n \log n)$ positioned squares and they have linear number of *runs*: maximal periodic subsegments (x is said to be *periodic* iff $period(x) \leq |x|/2$). The structure of runs in general strings is rather mysterious, and the structure of runs in Fibonacci words helps to understand this structure. In this sense Fibonacci words are very representative. A very good source for properties of these words is for example the book [10]. We rediscover/discover several known/unknown properties of Fibonacci words in a novel way: analyzing the automaton for the set of subwords. Let F_n be the n-th Fibonacci word, where

$F_0 = a,\ F_1 = ab,\ F_{n+1}\ =\ F_n \cdot F_{n-1}$

Denote by Φ_n the n-th Fibonacci number, where $|F_n| = \Phi_n$. Define also the infinite Fibonacci word $\mathcal{F}_\infty\ =\ \mathcal{F}_\infty(1, 2, 3, 4, \ldots)$, such that each F_n is a prefix of $\mathcal{F}_\infty$. Hence

* Research supported by the grants 4T11C04425 and CCR-0313219.

$\mathcal{F}_\infty \ = \ abaababaabaababaababaabaababaabaababaababaabaababaababaabaababaab\ldots$

By an *occurrence* of u in $\mathcal{F}_\infty$ we mean a position i such that $\mathcal{F}_\infty[i+1\ldots i+|u|] \ = u$. Denote by *first-ocuc*$(u)$ the first occurrence of u in $\mathcal{F}_\infty$, and by $occ(u)$ the set of all occurrences. The structure of lengths of paths in the dawg's of Fibonacci words is closely related to the Fibonacci number system. This system consists in representing a number as a sum of Fibonacci numbers, in such a way that no two consecutive Fibonacci numbers are used. The sum of zero number of integers equals zero. The corresponding representation of the number is called $\mathcal{Z}$-representation.

Theorem 1. [Zeckendorff Theorem, [11]]
Every nonnegative integer is uniquely represented in the Fibonacci number system. Every number $F_n \le k \le F_{n+1} - 1$ contains F_n as the largest term in its $\mathcal{Z}$-representation.

Define the *dual Fibonacci system*. In this system each positive integer x is represented as a sum of different Fibonacci numbers, however we require that if Φ_i is not taken then Φ_{i+1} is taken in the sum, whenever any Fibonacci number after Φ_i is taken. It follows directly from Zeckendorff's Theorem that:

Lemma 1. *Every integer $k > 0$ is uniquely represented in the dual Fibonacci number system.*

If X is a set of integers then define: $X \oplus j \ = \ \{x + j \ : \ x \in X\}$.
Denote by g_i (the i-th truncated Fibonacci word) the word F_i with the last two letters removed. Using the dawg's we show that for each nonempty subword u of $\mathcal{F}_\infty$ we have:

$$occ(u) \ = \ occ(g_i) \oplus \textit{first-ocuc}(u),$$

where g_i is the shortest truncated Fibonacci word containing u.

Let $\mathcal{Z}_n$ be the set of nonnegative integers which do not use Fibonacci numbers $\Phi_0, \Phi_1, \ldots, \Phi_{n-1}$ in their Fibonacci representation. It follows directly from the structure of the dawg that:

$$occ(g_{n+1}) \ = \ occ(F_n) \ = \ \mathcal{Z}_n \text{ for } n > 1 \text{ and } occ(F_1) = occ(F_2),\ occ(F_0) = \mathcal{Z}_1$$

The sorted set $\mathcal{Z}_k[0], \mathcal{Z}_k[1], \mathcal{Z}_k[2], \ldots$ is closely related to Fibonacci words, denote by $\mathcal{D}_k$ the displacement structure of $\mathcal{Z}_k$:

$$\mathcal{D}_k \ = \ (\mathcal{Z}_k[1] - \mathcal{Z}_k[0],\ \mathcal{Z}_k[2] - \mathcal{Z}_k[1],\ \mathcal{Z}_k[3] - \mathcal{Z}_k[2],\ \mathcal{Z}_k[4] - \mathcal{Z}_k[3],\ \ldots).$$

The following fact is very useful in the analysis of the structure of runs in $\mathcal{F}_\infty$. It shows that the dsiplacement sequence is isomorphic to $\mathcal{F}_\infty$.

Lemma 2. $\mathcal{D}_k \ = \ h_k(\mathcal{F}_\infty)$, *where* $h_k(a) = \Phi_k,\ h_k(b) = \Phi_{k-1}$.

It follows easily from the structure of the dawg's that every run in $\mathcal{F}_\infty$ (except aa, $(ab)^2$) is of the form $(F_i)^k g_{i-1}$, where $k \in \{2, 3\}$. In case of runs a similar analysis of the structure of runs of squares has been already done by Iliopoulos, Moore, and Smyth in [4]. However their proofs were syntactic, and here we present different graph-theoretic proofs, based on the structure of the dawg of $\mathcal{F}_\infty$ and on a natural number-theoretic interpretation of the sets of lengths of its paths.

2 The structure of subword graphs

We construct the infinite labelled graph $\mathcal{G}_\infty$, The nodes of $\mathcal{G}_\infty$ are all integers $i \geq 0$, the edges are constructed as follows:

$$(i-1) \stackrel{\mathcal{F}_\infty(i)}{\rightarrow} i, \text{ for } i > 0, \qquad \Phi_i - 2 \stackrel{b}{\rightarrow} \Phi_{i+1} - 1, \text{ for odd } i,$$

$$\text{and } \Phi_i - 2 \stackrel{a}{\rightarrow} \Phi_{i+1} - 1, \text{ for even } i.$$

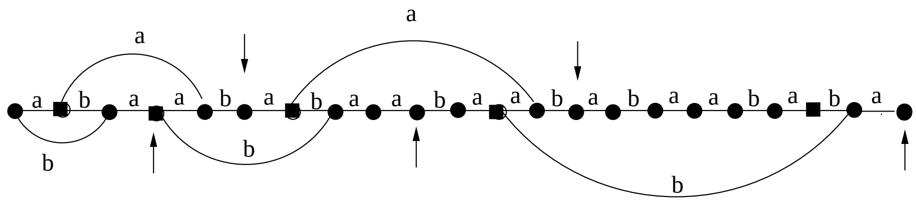

Fig. 1. The subword graph $dawg(F_6)$, the *fork nodes* (of outdegree 2) are drawn as squares. The arrows show the ends of prefixes which are Fibonacci words.

The graph $\mathcal{G}_\infty$ is, in a certain sense, a subword graph of the infinite Fibonacci word F_∞. The initial segments of this graph are dawg's of finite Fibonacci words. Denote by finite-paths($\mathcal{G}_\infty$) the set of all finite words spelled by the paths of $\mathcal{G}_\infty$ originating at 0, and by finite-subwords($\mathcal{F}_\infty$) the set of all finite sub-words of $\mathcal{F}_\infty$. The following fact follows from Theorem 6, which wil be proved later.

Theorem 2. *finite-paths*($\mathcal{G}_\infty$) $=$ *finite-subwords*($\mathcal{F}_\infty$).

We say that a path is an a-path if it is an infinite path in $\mathcal{G}_\infty$ which starts at 0, and chooses the edge labelled a whenever there is a choice. Similarly define b-path. Denote by a-path($\mathcal{G}_\infty$) the infinite word spelled by the a-path, similarly define b-path($\mathcal{G}_\infty$. The b-path($\mathcal{G}_\infty$) can be treated as the infinite lexicographically maximal *pseudo-suffix* of $\mathcal{F}_\infty$ (each prefix of b-path($\mathcal{G}_\infty$) is a prefix of maximal suffix of some finite Fibonacci word).

Theorem 3. $\quad$ *a-path*($\mathcal{G}_\infty$) $= a \cdot \mathcal{F}_\infty$, $\quad$ *b-path*($\mathcal{G}_\infty$) $= b \cdot \mathcal{F}_\infty$.

The edges of the form $(i, i+1)$ are called *main edges*. The suffixes of $\mathcal{F}_\infty$ are infinite words resulting by cutting off a finite prefix of $\mathcal{F}_\infty$.

Theorem 4.
a-path($\mathcal{G}_\infty$) *and b-path*($\mathcal{G}_\infty$) *are not suffixes of* $\mathcal{F}_\infty$.
The infinite string corresponding to a path π *of* $\mathcal{G}_\infty$ *is a suffix of* $\mathcal{F}_\infty$ *iff almost all edges of* π *(all but a finite number) are main edges.*

The nodes of outdegree greater than one are called fork nodes. We say that a path starting from 0 is a *fork-path* iff it ends at a fork node in $\mathcal{G}_\infty$. The next theorem follows from the structure of the compacted infinite dawg $\mathcal{G}_\infty$. However we introduce later the compacted dawg's in terms of finite words.

Theorem 5. *For each* $k > 1$ *there is exactly one fork-path of length* k *in* $\mathcal{G}_\infty$. *This fork-path corresponds to the representation of* k *in the dual Fibonacci number system.*
For each $\Phi_{n+1} - 2 < k \leq \Phi_{n+2} - 2$ *there is a path of length* k *from the source to the fork node* $\Phi_{n+2} - 2$.

Proof. It is easy to see that each path of total length k from the source to a fork node in the compacted version of $\mathcal{G}_\infty$, see Figure 4, corresponds to a representation of a number k in the dual Fibonacci system, the example of representing $k = 60$ is shown in Figure 4.

If we have paths of length $k-1$ then only a path ending at a fork node generates two paths of length k. The theorem and Lemma 1 implies directly in a novel way the following well known Sturmian property of $\mathcal{F}_\infty$.

Corollary 1. *There are exactly* $k + 1$ *different subwords of length* k *in* $\mathcal{F}_\infty$.

Let $\mathcal{G}_n$ be the subgraph of $\mathcal{G}_\infty$ induced by the nodes $[0 \ldots \Phi_n]$, see Figure 1. Denote by $dawg(w)$ the acyclic directed subword graph of a word w, see for example [2, 7, 10] for the definition. We assume that the nodes on the *main branch* of such a graph are consecutive integers starting with 0.

Theorem 6. *For each* $n > 1$ $dawg(F_n) = \mathcal{G}_n$ *and paths*($\mathcal{G}_n$) $=$ *suffixes*(F_n).

Proof. The thesis follows from the on-line construction of $dawg(F_n)$, see [2]. It is enough to show that no extra nodes outside the main branch are created. If $dwag(F_n) = \mathcal{G}_n$ then the next $|F_{n-1}| - 2$ symbols do not create new nodes or new edges since g_n is a prefix and suffix of g_{n+1}, which consequently has the period $|F_{n-1}|$. One extra edge is created from $|F_n| - 2$ to $|F_{n+1} - 1|$ because the next read symbol terminates the period $|F_{n-1}|$. We omit the details.

We refer the reader to [2] for the definition of the critical factorization point. The starting position of a lexicographically maximal suffix, maximized over all possible orders of the alphabet, is the critical factorization point. This implies the following fact:

Theorem 7. $\Phi_n - \min\{\ |a\text{-}path(\mathcal{G}_n)|,\ |b\text{-}path(\mathcal{G}_n)|\ \}$ *is the critical factorization point of the n-th Fibonacci word.*

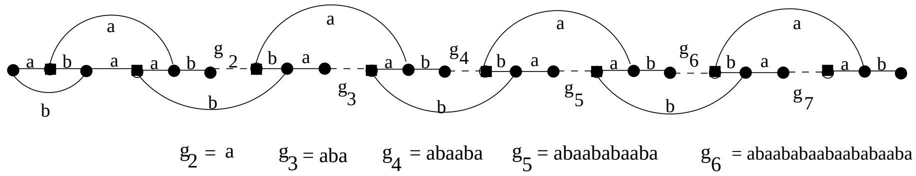

Fig. 2. The *structure* of $dawg(F_9)$, of 12th Fibonacci word (of length 89). The dashed edges correspond to chains.

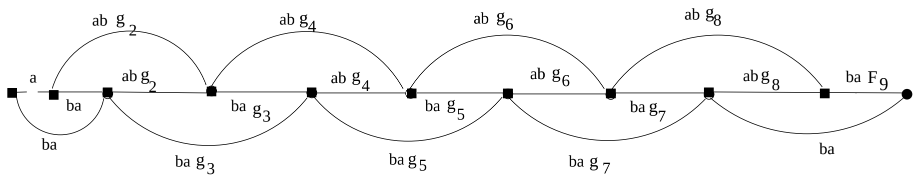

Fig. 3. The power of compaction: $cdawg(F_{11})$ of the Fibonacci word of length 233. Observe that all labels (but one) are reverses of Fibonacci words.

This gives alternative proof, see [3], of the following fact.

Corollary 2. *$\Phi_{n-1} - 1$ is a critical factorization point of F_n.*

$dawg(F_n)$ can be compactly described in $O(n)$ space, see Figure 2 for the first compaction, in which each chain (a sequence of nodes of indegree and outdegree one) is represented by a single edge. We can further compact $dawg(F_n)$. Let us remove all nodes except fork nodes, the source and the sink. Call remaining nods essential. Then for each edge outgoing from an essential node replace it by an edge going to the next essential node, with label representing the word "spelled" by the compressed path, see Figure 3. The resulting compacted subword graph is denoted by $cdawg(F_n)$.
By $O(1)$ space we mean constant number of nonnegative integers not greater than n.

Theorem 8.
We can test if a word w is a subword of a Fibonacci word in time $O(|w|)$ and $O(1)$ space.

Proof. It is easy to see that we can test if a specified subword of w is a Fibonacci word in linear time and $O(1)$ space. Then we can traverse $\mathcal{G}$ without remembering it explicitly. In some places we have to test if a subword of w is a Fibonacci word.

Define $fin(u) \;=\; occ(u) \oplus |u|$ and *first-fin*$(u) \;=\; \min(fin(u))$.

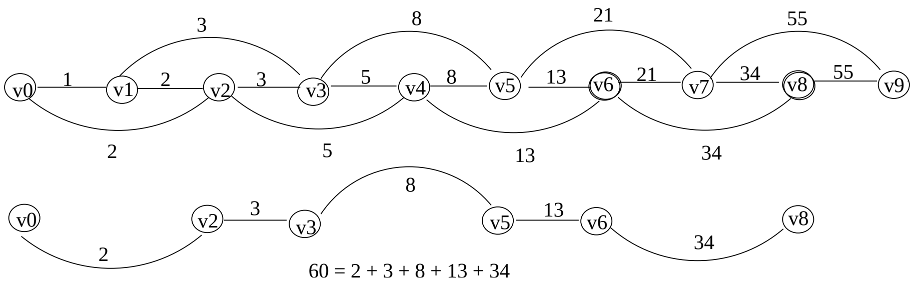

Fig. 4. We consider only fork nodes and put the lengths of the edges as the lengths of compacted paths. The representation of $k = 60$ in the dual Fibonacci system corresponds to a path $(v_0,\ v_2,\ v_3,\ v_5,\ v_6,\ v_8)$. It illustrates the fact that for each k exactly one word corresponds to a path from the source to a fork node.

Lemma 3.
(A) *For each pair of nonempty words u, w we have:*

$$\textit{first-fin}(u) = \textit{first-fin}(w) \;\Leftrightarrow\; occ(u) = occ(w).$$

(B) *For each nonempty subword u of $\mathcal{F}_\infty$ we have*
$occ(u) \;=\; occ(g_i) \oplus \textit{first-ocuc}(u)$, where g_i is the shortest truncated Fibonacci word containing u.

Example.
The shortest truncated Fibonacci word containing aa, as well as $F_3 = abaab$ is $g_4 = abaaba$. We have $occ(F_3) = occ(g_4) = \{0, 5, 8, 13, 18, 21, 26, 29, \ldots\}$, and $\textit{first-ocuc}(aa) = 2$, hence

$$occ(aa) \;=\; occ(F_3) \oplus 2 \;=\; occ(g_4) \oplus 2 \;=\; \{2, 7, 10, 15, 20, 23, 28, 31, \ldots\}.$$

The structure of the graph $\mathcal{G}$ implies easily several number-theoretic properties of the Fibonacci words. It follows from Lemma 3 and the structure of the graph $\mathcal{G}$, see Figure 4, that:

Theorem 9.

1. *$occ(g_{n+1}) \;=\; occ(F_n) \;=\; \mathcal{Z}_n$ for $n > 1$*
2. *$occ(F_1) = occ(F_2)$, $occ(F_0) = \mathcal{Z}_1$.*
3. *For each subword $u \notin \{F_0, F_1\}$ of $\mathcal{F}_\infty$ we have $occ(u) \;=\; \mathcal{Z}_i \oplus \textit{first-ocuc}(u)$, where g_i is the smallest truncated Fibonacci word containing u as a subword.*

Proof. The subword u *"moves"* to the right by starting at $\textit{first-ocuc}(u)$ in $\mathcal{G}$ and making shortcuts. Each shortcut corresponds to taking a Fibonacci number, no two consecutive Fibonacci numbers are taken.

We investigate also the structure of the set

$FIN(k) \;=\; \{\textit{first-fin}(u) \;:\; u \text{ is of size } k\ \}$

The structure of this set easily follows from the way how paths of length $k-1$ are extended into paths of length k. Only fork nodes $i \in FIN(k-1)$ generate two elements of $FIN(k)$, each other node i in $FIN(k-1)$ generates a single element $i+1$ in $FIN(k)$, see Figure 5. We have:

$$FIN(k+1) \;=\; FIN(k) \oplus 1 \cup \{\Phi_{i+1}-1\} \text{ where } \Phi_i - 2 \in FIN(k)$$

This implies directly the following fact.

Lemma 4.
The set $FIN(k)$ consists of a single interval or of two disjoint intervals.
$FIN(\Phi_n - 1) \;=\; [\Phi_n - 1 \ldots 2\cdot\Phi_n - 1]$;

We say that a subword w of $\mathcal{F}_\infty$ is a right special subword, iff wa and wb are subwords of $\mathcal{F}_\infty$. Such subwords are responsible for the increase of the number of subwords with respect to their length. These are the words corresponding to paths to fork nodes, they are considered for example in [1]. It is easy to see from the structure of $\mathcal{G}$ that right special subwords are exactly suffixes of g_i's. On the other hand each suffix of g_i is a reverse of a pefix of $\mathcal{F}_\infty$. Let w^R denote the reverse of w. We gave a new proof of another property of $\mathcal{F}_\infty$:
a word w is a right special factor of $\mathcal{F}_\infty$ iff w^R is a prefix of $\mathcal{F}_\infty$.

3 The structure of runs in Fibonacci words

We say that a run w is a p-run iff $period(w) = p$. The run is short if $|w| < 3\cdot period(w)$. The structure of runs has been already investigated in [8, 4]. Every occurrence of a subword in $\mathcal{F}_\infty$ implies an occurrence of some word g_i starting at the same position. Hence the runs correspond to adjacent occurrences or overlaps of words g_i. Consequently we have the following fact.

Lemma 5. [8] *Every run of $\mathcal{F}_\infty$ is of one of two types:*
(Short runs) $w = F_k \cdot F_k \cdot g_{k-1}$, *or* **(Long runs)** $w = F_k \cdot F_k \cdot F_k \cdot g_{k-1}$.

Denote by $rep(x)$ the repetition order of the string (finite or infinite) x as
$rep(x) \;=\; \sup\,\{|w|/period(w) \;:\; w \in \text{finite-subwords(x)}\}$.

The maximal repetitions correspond to long runs in $\mathcal{F}_\infty$. This implies the following fact (already shown in [8]).

Corollary 3. [8] $rep(\mathcal{F}_\infty) = 2+\phi$, *where* $\phi = \frac{1+\sqrt{5}}{2}$ *is the golden ratio.*

It follows from the structure of runs that there is no subword aaa in $\mathcal{F}_\infty$. Using the displacement sequence $\mathcal{D}_k$, due to its recursive Fibonacci-like structure, we can easily show the following (already shown using different methods) properties of Fibonacci words.

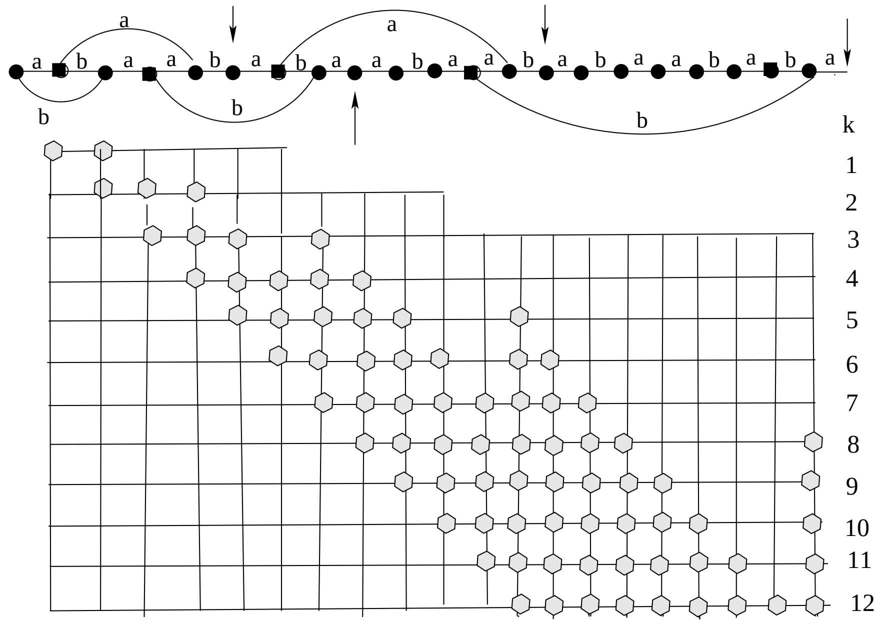

Fig. 5. The structure of nodes of G which are endpoints of all $k+1$ different strings of length k. The end-positions of Fibonacci prefixes are indicated by vertical arrows.

Corollary 4. *There are no subwords in $\mathcal{F}_\infty$ of type x^4, where x is nonempty. For $n > k$, the number of occurrences of F_k in F_n is $F_{n-k} - odd(n-k)$, where $odd(x) = 1$ if x is an odd integer, and $odd(x) = 0$ otherwise.*

All runs correspond to occurrences of g_i's. However $\mathcal{Z}_i$ is the set of all occurrences of g_i. The crucial role in understanding the structure of runs in F_n plays the Displacement Lemma (Lemma 2). We know that the displacement sequence is isomorphic to Fibonacci sequence, hence we can easily compute number of different types of runs by computing numbers of a's and b's in prefix segments of $\mathcal{F}_\infty$. Using Lemma 2 we can describe the structure of runs in F_n, see Figure 6.

Theorem 10. [**Structure of Runs**] *The Fibonacci word F_n has: F_{n-3} Φ_0-runs; $F_{n-4} - 1$ Φ_1-runs, and F_{n-k-2} Φ_k-runs for $2 \le k \le n-2$.*

This gives alternative (compared with [6]) proof for the number of all runs.

Corollary 5. [6] *F_n has $2 \cdot F_{n-2} - 3$ runs.*

We say that a square xx is primitive iff x is a primitive word, similarly define primitive cubes. The run $F_iF_ig_{i-1}$ contains $|g_{i-1}|$ primitive squares and

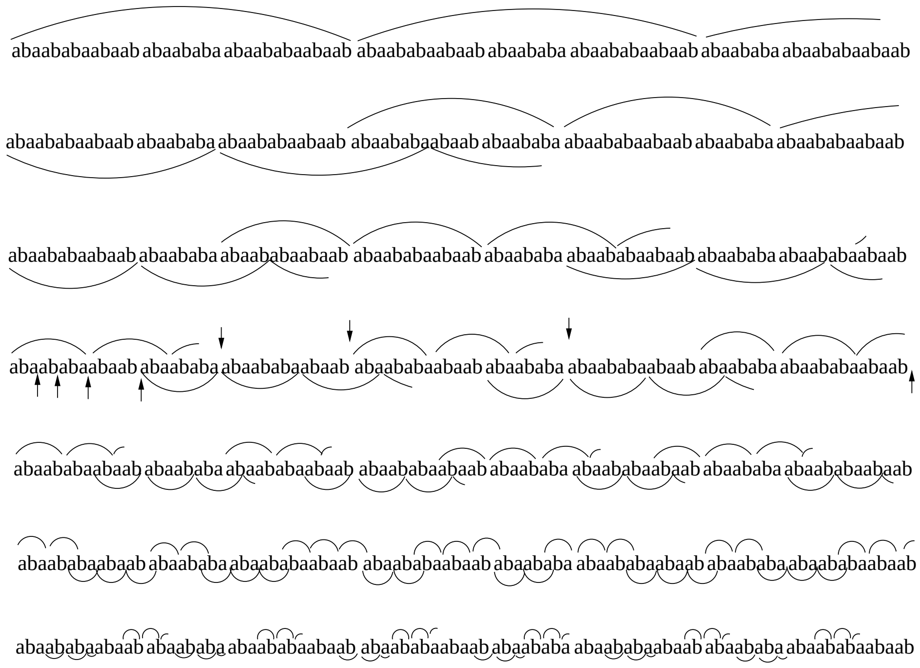

Fig. 6. The structure of *runs* in the Fibonacci word F_{11}. The arrows show endpoints of Fibonacci prefixes. F_{11} has 65 runs. The 21 1-runs of aa are not shown in the figure. The runs are distributed as follows: there are 12 2-runs, 13 3-runs, 8 5-runs, 5 8-runs, 3 13-runs, 2 21-runs and 1 34-run.

$F_iF_iF_ig_{i-1}$ contains $\Phi_i + |g_{i-1}|$ primitive squares. The short runs correspond to bab in $\mathcal{F}_\infty$ and long runs correspond to aa in $\mathcal{F}_\infty$. Due to the Fibonacci-like structure of displacement sequences (Lemma 2) and Theorem 6 we can calculate (in a new easier way) the number of all positioned primitive squares and all primitive positioned cubes in finite Fibonacci words.

4 The structure of suffix trees

The suffix tree T_n of F_n is the tree of all paths of $cdawg(F_n)$. The structure of this tree and the way how T_n evolves into T_{n+1} follows from the structure and evolution of compacted dawg's, see Figure 7 and Figure 8.

A terminal edge is an edge leading to a leaf. The suffix trees of Fibonacci words grow at their leaves, by changing the terminal edges in a very simple regular way.

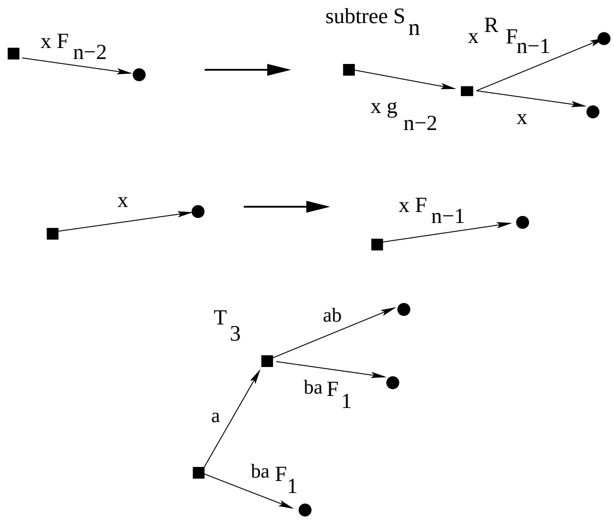

Fig. 7. The suffix tree T_3 and the general rules to generate T_{n+1} from T_n. The word $x \in \{ab,\ ba\}$.

Theorem 11. *For $n > 2$ the suffix tree T_n of F_n has Φ_{n-1} leaves and Φ_{n-2} internal nodes. Let x be the last two symbols of F_{n+1}. T_n evolves into T_{n+1} in the following way:*

- **(long edges)** *each terminal edge (u, v) with label xF_{n-2} is transformed into the subtree isomorphic to S_n, two end symbols are cut off from the label of (u, v), and two edges originated at v are created , with labels $x^R F_{n-1}$ and x,*
- **(short edges)** *each terminal edge with label x changes its labels to $x \cdot F_{n-1}$.*

We know precisely how the suffix trees grow. The sum of lengths of edges of the suffix tree is the number of different subwords. We have Φ_{n-3} short edges, each of them grows by Φ_{n-1}, and Φ_{n-2} long edges, each grows by $\Phi_{n-1} + 2$. This gives easily a simple recurrence and a new suffix-tree oriented proof for the known formula of the number $Sub(n)$ of different subwords of F_n.

Corollary 6. *For $n > 2$ we have:*

$$Sub(n+1) \ = \ Sub(n) + \Phi_{n-3} \cdot \Phi_{n-1} + \Phi_{n-2} \cdot (\Phi_{n-1} + 2)$$
$$Sub(n) \ = \ \Phi_{n-1}\Phi_{n-2} + 2 \cdot \Phi_{n-1} - 1$$

We say that two labelled trees are structurally isomorphic iff they are isomorphic as unordered trees in graph-theoretic sense, disregarding the labels. The following fact also follows from the structure of cdawg's.

Theorem 12. [Fibonacci-like structure of suffix trees of Fibonacci words] *For $n > 4$ the two subtrees rooted at the sons of the root of the suffix tree $T(F_n)$ are structurally isomorphic to the suffix trees $T(F_{n-1}$ and $T(F_{n-2})$, respectively.*

Acknowledgment I thank Łukasz Mikulski [9] for helpful comments related to Lemma 4. He has written an independent and more formal proof of this lemma. The subword graphs for general Sturmian words have been already considered in [5]. There is a huge literature on Sturmian words (each Fibonacci word is a Sturmian one).

References

1. J. Berstel, J. Karhumaki, Combinatorics on words - a tutorial. Bull. EATCS 79 (2003), pp 178-228.
2. Crochemore, M., and Rytter, W., Jewels of stringology: text algorithms, World Scientific (2003)
3. T. Harju, D. Nowotka, On the density of critical factorizations, ITA 36(3): 315-327 (2002)
4. C. Iliopoulos, D. Moore, W.F. Smyth, A Characterization of the Squares in a Fibonacci String. Theor. Comput. Sci. 172(1-2): 281-291 (1997)
5. C., Mignosi F., Shallit J., Venturini I. (2004). Sturmian Graphs and a Conjecture of Moser. DLT'04,Lecture Notes in Computer Science 3340 (2004) 175-187.
6. R.Kolpakov, G.Kucerov, Finding Maximal Repetitions in a Word in Linear Time. FOCS 1999: 596-604
7. Ch. Charras, T. Lecroq, Handbook of exact string matching, King's College Publications 2004
8. F. Mignosi, G. Pirielo, Repetitions in the Fibonacci inifinite words, RAIRO Theor. Inf. Appl. 26, 194-204 (1992)
9. Ł. Mikulski, Personal communication, 2005
10. B. Smyth, Computing patterns in strings, Addison Wesley 2003
11. E. Zeckendorff, Representation des nombres natureles par une somme de Fibonacci ou de nombres de Lucas, Bulletin de la Societe Royale des Sciences de Liege 41 (1972) 179-182

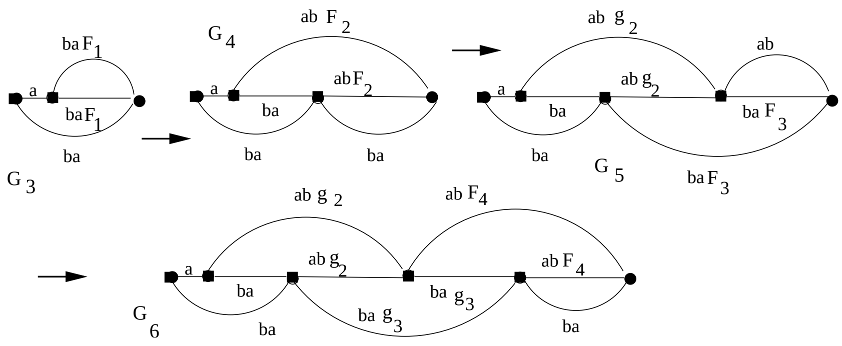

Fig. 8. The evolution of the compacted graph $cdawg(F_6)$.

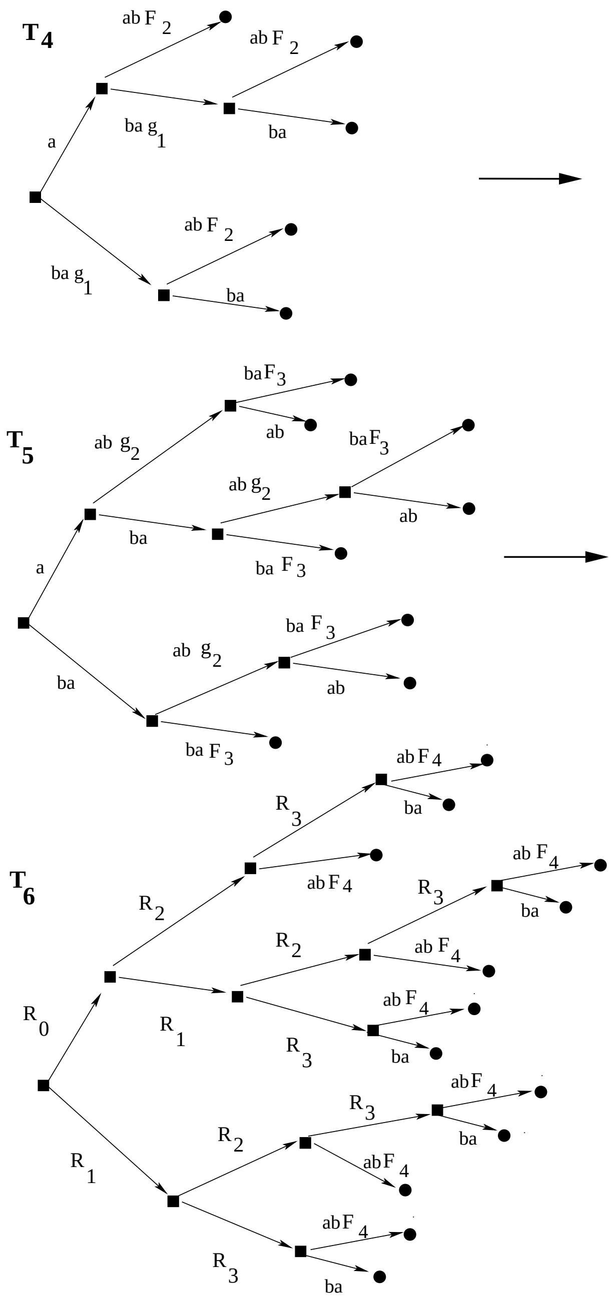

Fig. 9. The evolution of the suffix tree $T_6 = T(F_6)$. Observe that g_1 is the empy string and that the labels $ab\ g_i$ (for even i) and $ba\ g_j$ (for odd j) are reverses of Fibonacci words, denote $R_k = F_k^R$. We can obtain *in the limit* an infinite suffix tree $\mathcal{T}_\infty$ of $\mathcal{F}_\infty$.